A Measurement Based Memory Performance Evaluation of Streaming Media Servers

Garba Isa Yau and Abdul Waheed
E-mail: {dygarba, awaheed}@ccse.kfupm.edu.sa
Computer Engineering Department
King Fahd University of Petroleum and Minerals
Dhahran 31261, Saudi Arabia

Abstract

While a number of studies have focused on storage subsystem performance, in general very few studies have explicitly focused on the memory subsystem performance of streaming media servers. We carried out measurement-based study of the memory performance of two leading streaming media servers: Darwin streaming server and Windows media server. Our goal is to determine the specific conditions under which on-chip cache or main memory becomes major bottleneck on the performance of these streaming media servers. Our measurement-based analysis indicates that with large number of client requests and high encoding rate (300kbps), the memory performance degrades significantly, leading to excessive number of cache misses and page faults which leads to throughput degradation and client timeout. Windows media server exhibits better cache performance and reports higher throughput. However, Darwin streaming server has lower page fault rate at 300kbps encoding rate and multiple stream distribution.

Keywords: streaming media, encoding rate, cache miss, page fault, video on demand.

1. Introduction

Streaming media is a popular paradigm on the internet. The popularity is increasing in terms of both large number of users as well as the variety of applications. Audio and video clips can be digitized, encoded, and forwarded to the end user in smaller media object for viewing. To view these media objects special software tools are used to decompress and render them on the screen. Unlike the concept of downloadable media, streaming media technology enables a user to experience a multimedia presentation on the fly, while it is being downloaded from the Internet. For exhaustive description of the architecture and working mechanism of streaming media servers, the reader is referred [1,2,3].

Because of the stringent timing and quality-of-service requirements, high-bandwidth demands, and the CPU and memory intensive characteristics of these applications, performance of the server hardware is critical for efficient delivery of high quality multimedia content.

We present an experimental study of the memory performance of streaming media servers. We obtained low-level details of server performance for a number of configurations. We obtained measurements for cache misses and page faults using two leading streaming media servers: Darwin streaming server and Windows media server.

The rest of this paper is organized as follows. In section 2, we review some related work and outline our motivation for this study, while in section 3 we briefly discuss some characteristics of continuous media data. Sections 4 and 5 discuss our experimental setup and results. In section 6, we report the effect of operating system on server performance while conclusion and future direction of this research are outlined in section 7.

2. Related Work and Motivation

Growing deployment and use of streaming media servers is already drawing the attention of researchers. Performance of a streaming server is a key factor contributing to the quality of the multimedia content for the end-users. Shenoy et al [4] highlighted some fundamental issues arising in multimedia server design. Technical design challenges such as storage and retrieval of multiresolution data, scalability and management were presented. Sohn et al [5] looked at the performance of a small-scale VOD server. They conducted a measurement-based study in which they outlined the predictability of the real-time scheduler and the performance of the VOD server. A significant amount of work is reported in the literature on the disk storage performance for streaming media servers. Due to large volumes of video and other multimedia files, storage and retrieval techniques play an important role in the performance of the server too. A storage hierarchy to design a low-cost cache for a movie on demand (MOD) server was proposed in [6]. The hierarchy consists of a disk, which stores the popular movies, and a small amount of RAM buffers that store only portions of the movies. Due to low cost of disks, the cost of a MOD server based on the proposed architecture is substantially less than one in which the entire movie is loaded into RAM.

Some studies of multimedia servers pay attention to I/O subsystems due to the high throughput demand of the servers. In fact, streaming media servers are often I/O bound. A study by Batatia et al [7] focuses on the design of an I/O subsystem for a continuous media server. They propose several improved architectures based on an existing device: Intel i960RP I/O processor, and evaluate their performance. They report that utilization of the I/O processor solved the main memory bottleneck problem but created a new bottleneck in i960RP memory. I/O performance in multimedia servers has also been investigated using simulation [8]. Various I/O issues in multimedia systems have been discussed in [9].

Our literature search reveals that researchers did not pay much attention to the memory performance of the streaming server itself. An exception is the study conducted by Sohoni et al [10] which rather study the memory system performance of multimedia applications at the client end. Despite poor temporal locality, at the client's application end, high cache hit ratio is reported which is attributed to factors like block partitioning algorithm employed, significant data reuse and excellent spatial locality of continuous multimedia data. However, this study does not address server performance and its high throughput demand.

There has been tremendous progress in microprocessor technology, which leads to high speed CPUs. Also, advances in memory and magnetic disk technology have lead to improvement in memory density and magnetic disk density much more than access and cycle times. Density of semiconductor DRAM increases by 60% per year, quadrupling in three years, but cycle time has improved very slowly, decreasing by about one-third in 10 years. In a similar fashion, magnetic disk density has been improving by about 50% per year, almost quadrupling in three years. Access time has improved by only one-third in 10 years [11]. It is obvious that memory and disk performance can limit the performance of a busy streaming media server that serves highly popular compressed audio/video contents.

3. Characteristics of Continuous Data

In our attempt to highlight the performance of streaming media servers, discussing the nature of the streaming content will be essential in laying out the framework on which our performance study is based. However, due to lack of space, we will not attempt an exhaustive discussion on that. Streaming media content falls into the category of continuous media. Continuous media are characterized by a timing relationship between the server and the client [12].

Considering continuous media in the context of the memory hierarchy, we are confronted with mixed factors affecting performance; some positively while others negatively. At the server end, continuous media has excellent spatial data locality but a very poor temporal locality since there is almost no data reuse. The large amounts of data flowing through the CPU and memory subsystem significantly decreases overall cache hit ratios since they cannot fit into cache, leading to degradation of performance due to slow memory accesses [12]. This is contrary to what happens at the application end where high cache hit rates occur for multimedia applications because of a number of factors like use of block partitioning algorithms to the input data, working on small blocks of data that easily fit into cache and high data reuse [10].

4. Experimental Configuration

4.1 Test bed

Our experimental test bed comprises of a dual boot server machine that hosts the streaming servers under appropriate operating system and six client machines running streaming load simulators. The setup consists of a closed-LAN with a Cisco 1 Gbps multilayer switch (catalyst 3550). The streaming media servers are executed on a PC with Pentium IV 2.0 GHz, 256 MB SDRAM, single 40 GB EIDE hard drive (Western Digital WD400) and 3Com 1 Gbps Ethernet NIC. Server was booted with Windows 2000 server to test Windows media server. The same server was booted with Red Hat Linux 7.2 (kernel 2.4.10) to test Darwin streaming media server. The clients run on PCs, each comprising of a Pentium III 300 MHz, 96 MB RAM and 100Mbps NIC. The load simulator can generate a large number of client requests from a single client computer. Figure 1 illustrates our experimental test bed.

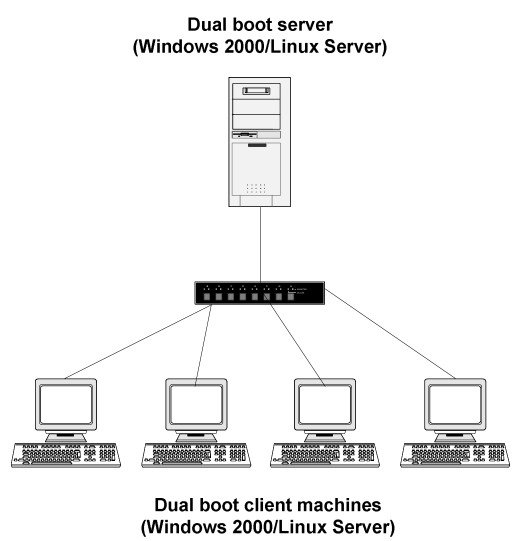

Figure 1: Experimental test bed

4.2 Metrics and factors

Our metrics of interest are:

- cache misses (level 1 and 2 cache)
- page fault rate
- server aggregate throughput
- server CPU utilization

To observe our metrics, we vary the following factors:

- number of streams (streaming clients)
- media encoding rate (56kbps and 300kbps)
- stream distribution (unique or multiple media)

We setup a video-on-demand scenario where clients request for stored compressed video streams from the server.

4.3 Tools

We collect measurements for our metrics using a number of software tools that run on the server machine in a non-intrusive way. Some tools run on both platforms (Windows and Linux) while others run only on one platform. However, for platform specific tools, we ensure that such tools exhibit very similar overhead (generally minimally intrusive) on the operating system in question. The following tools were used:

- Streaming Load Tool - for Windows media server we use Microsoft streaming load simulator while for Darwin streaming media server, we use streaming load simulator. Both simulators operate in similar manner by making requests for media objects through launching a large number of clients.
- Intel VTune performance analyzer (6.1): performance and profiling tool that we use to access CPU on-chip performance counters. It is available for both Windows and Linux
- Windows 2000 'performance': a Windows platform performance tool.
- Netstat: a tool for measuring bandwidth and observing connection status. It is available both on Windows and Linux platforms.
- Linux tools: vmstat, iostat and sar.

5. Experimental Results and Discussion

In this section, we present the results of our measurement based performance study. We report measurements obtained using tools described in section 4.

5.1 Cache Performance

Figures 2 and 3 show the L1 cache behavior under different configurations of clients, encoding rates and stream distribution. Measurements for both servers are reported. Both figures show increase in cache misses as the number of clients increases. Thought not to a large extend, the stream distribution and encoding rate also affect the miss rate. We observe the worse case cache misses in both L1 and L2 when there is a large number of clients requesting multiple streams at 300kbps encoding rate. For this case, we started observing clients being refused connection by the server, which eventually makes the client to time out. We show L2 cache behavior for 300kbps encoding in Figure 4. For all these cases, Windows media server exhibits lower L1 and L2 cache misses. It becomes obvious that as a result of large working data set and lack of data reuse by the server, thrashing occurs in the CPU cache which renders the cache ineffective for this type of application.

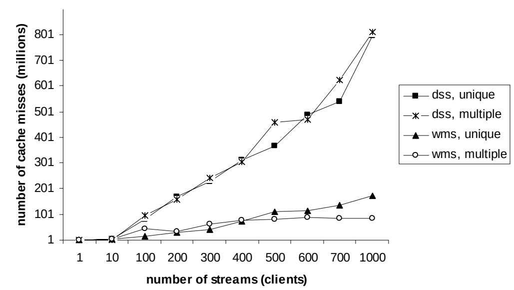

Figure 2: L1 cache misses for 56kbps encoding rate

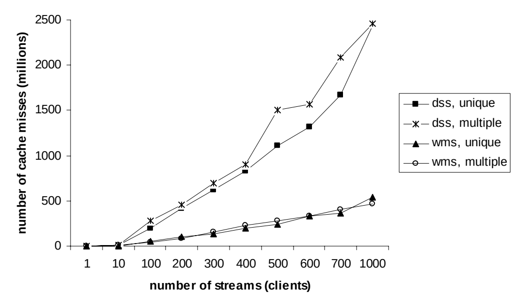

Figure 3: L1 cache misses for 300kbps encoding rate

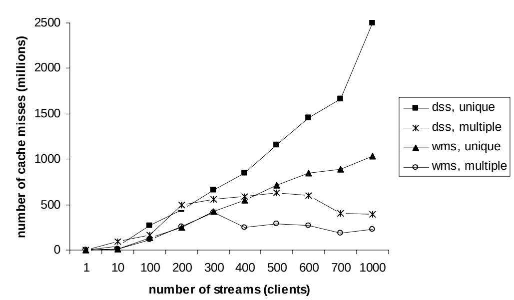

Figure 4: L2 cache misses for 300kbps encoding rate

5.2 Memory Performance

We consider main memory performance in terms of page fault rate. Any data referenced in the memory that is not available would have to be fetched from the disk, an expensive and slow process. Disk access is very slow especially if there are lots of disk references within a short duration. This is what we observe when we have clients requesting different media files. Figure 5 shows the page fault rate for clients requesting multiple streams at 300kbps. For cases where a unique stream is requested, this can be served from the memory and we observe a fairly constant page fault rate, which indicates low disk activity. However, as the clients request multiple streams, the page fault rate steadily increases with the number of clients. This is due to larger volume of data that have to be fetched from the disk into the main memory as the number of requesting streams grow larger.

The consequence of this high page fault rate is clients' timeout. It is intuitive that as more disk activity is involved, the server responds to clients' request very slowly leading to some of the clients to timeout after waiting for response for a long period of time.

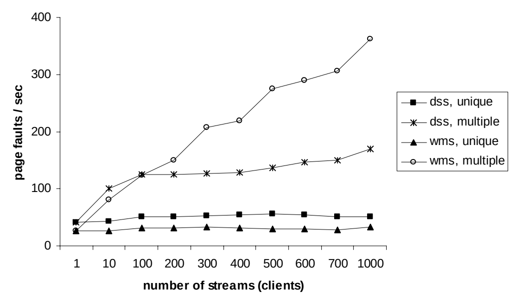

Figure 5: page fault rate for 300kbps encoding rate

5.3 Throughput and CPU Utilization

We show the throughput for 300kbps encoding rate and the corresponding CPU utilization. In Figure 6, the throughput for unique streams increases with the number of clients for both Darwin streaming server and Windows media server. However, for multiple streams, the throughput hardly increases beyond 200 streams. In fact, at 400 clients, we started observing clients timeout in Windows media server while we observe timeout in Darwin streaming server at 1000 streams. In all these cases, Windows media server has higher throughput. The corresponding CPU utilization for this configuration is shown in Figure 7. There is increase in CPU utilization as the number of clients increases. For unique stream access, the CPU utilization increases all the way up to 1000 streams. CPU utilization begins to drop when the number of clients timing out increases, thereby reducing the effective number of clients requesting media streams. Although this appears strange in the case of Darwin streaming server in which clients timeout begins at 1000 multiple streams, we still attribute the drop in CPU utilization after 300 multiple streams to low volume of data served to the clients.

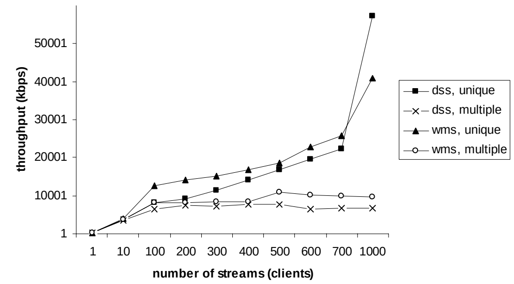

Figure 6: Server Throughput for 300kbps encoding rate

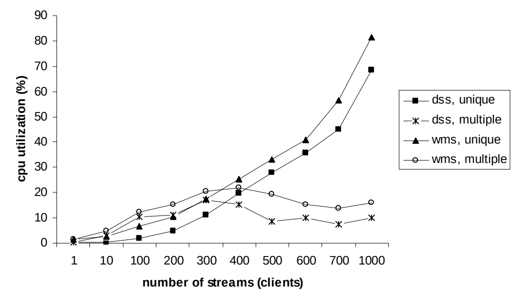

Figure 7: Server CPU utilization for 300kbps encoding rate

6. Operating System Impact on Performance

It is not our goal to benchmark the performance of Darwin streaming media server and Windows media server. However, we suspect that since the media servers were run on different platforms, there might be differences in memory behavior due to the operating system. To observe this effect, we made a quick memory subsystem assessment. ECT (extended copy transfer) memperf [13] is a method to characterize the performance of memory systems. It captures two aspects of the memory hierarchy: its behavior with temporal locality by varying the working set size (block size) and the spatial locality by varying the access pattern (strides). The calculated value is the transfer bandwidth (for a large amount of data). We present the extended copy transfer characterization for load sum test. The load test measures the memory load performance for all the block-sizes and access patterns. Figure 8 shows our memperf microbenchmark result for Linux and Windows running on the same hardware. Both operating systems show similar memory performance, with the memory bandwidth decreasing as the block size increases; that is not fitting into cache. The worst case is when the block size is beyond 512kB which is the size of the level 2 cache. We run test only for stride = 1, representing contiguous data, as that is the case with streaming media data. Based on these results, we conclude that difference in memory performance by the streaming servers is inherent in the servers themselves and not their host operating system.

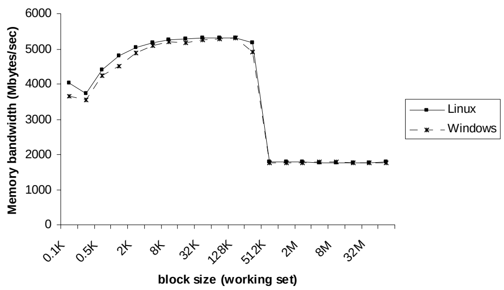

Figure 8: Extended copy transfer characterization (stride = 1)

7. Conclusions and Future Work

In this paper, we have presented the measurement-based evaluation of memory performance of two streaming media servers. We conducted experiments on Darwin streaming media server and Windows media server under identical workload conditions but different operating systems. Our measurements show that cache and memory subsystem performs worst when there is a large number of clients requesting multiple media streams at 300kbps encoding rate. Both servers exhibit similar characteristics in cache/memory performance under identical workload. The large number of cache misses and page faults leads to significant wastage in CPU cycles and high memory latency, hence a bottleneck on performance.

Since it is obvious from this study that memory is a major bottleneck in the performance of streaming servers, we are looking into ways to alleviate this bottleneck. Streaming media servers could be designed to take advantage of the excellent spatial locality in continuous media by incorporating techniques like lookahead prefetching and demand prefetching. Such techniques can improve cache hit rate and reduce page fault rates. Our work continues to explore these issues.

Acknowledgement The authors would like to thank King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia for supporting this research effort.

References

[1] H. Pang, B. Jose and M.S. Krishnan, "Resource scheduling in a high-performance multimedia server," IEEE Transactions on Knowledge and Data Engineering, vol. 11, pp. 303{320, March/April 1999.

[2] G. J. Conklin, G. S. Greenbaum, K. O. Lillevold, A. F. Lippman and Y. A. Reznik, "Video coding for streaming media delivery on the internet," IEEE Transactions on Circuit and Systems for Video Technology, vol. 11, pp. 269-281, March 2001.

[3] D. Wu, Y. Thomas, W. Zhu, Y. Zhang and J. M. Peha, "Streaming video over the internet: Approaches and directions," IEEE, Transactions on circuit and systems for video technology, vol. 11, pp. 282-300, March 2001.

[4] P. J. Shenoy, P. Goyal and H. Vin, "Issues in multimedia server design," ACM Computing Surveys, vol. 27, December 1995.

[5] J. M. Sohn, G. Y. Kim and T. G. Kim, "Performance measurements of a small-scale vod server based on the unix," The Third IEEE Symposium on Computers and Communications ISCC'98, June 1998.

[6] B. Ozden, A. Biliris, R. Rastogi and A. Silberschatz, "A disk-based storage architecture for movie on demand servers," Information Systems, vol. 20, no. 6, p. 465, 1995.

[7] M. Weeks, H. Batatia and R. Sotudeh, "Improved multimedia server i/o subsystems," Euromicro98, 24th Conference Proceedings, 1998.

[8] M. Weeks and C. Bailey, "Continuous discrete-event simulation of a continuous-media server I/O subsystems,"Euromicro 2000, Workshop on Multimedia and Telecommunications, September 2000.

[9] A. L. Reddy and J. Wyllie, "IO issues in a multimedia system," IEEE Computer, vol. 27, pp. 69-74, March 1994.

[10] S. Sohoni, R. Min, Z. Xu and Y. Hu, "A study of memory system performance of multimedia applications," Proceedings of the 2001 ACM SIGMETRICS international conferenceon Measurement and modeling of computer systems, pp. 206-215, 2001.

[11] J. L. Hennessy and D. A Patterson, Computer Architecture: A Quantitative Approach. Morgan Kaufmann Publishers, Inc., 1996.

[12] H. Schulzrine, "Operating system issues for continuous media," Multimedia Systems, vol. 4, pp. 269-280, October 1996.

[13] C. Kurmann and T. Stricker, "Characterizing memory system performance for local and remote accesses in high end smps, low end smps and clusters of smps," 7th Workshop on Scalable Memory Multiprocessors held in conjuction with the 25th Annual International Symposium on Computer Architecture ISCA98, June 1998.